Math. Program., Ser. A (2013) 141:527–547
DOI 10.1007/s10107-012-0565-4

FULL LENGTH PAPER

A 3-approximation algorithm for the facility location problem with uniform capacities

Ankit Aggarwal · Anand Louis · Manisha Bansal · Naveen Garg · Neelima Gupta · Shubham Gupta · Surabhi Jain

Received: 28 July 2010 / Accepted: 25 April 2012 / Published online: 9 June 2012

Abstract We consider the facility location problem where each facility can serve at most U clients. We analyze a local search algorithm for this problem which uses only the operations of add, delete and swap and prove that any locally optimum solution is no more than 3 times the global optimum. This improves on a result of Chudak and Williamson who proved an approximation ratio of $3 + 2\sqrt{2}$ for the same algorithm. We also provide an example which shows that any local search algorithm which uses only these three operations cannot achieve an approximation guarantee better than 3.

Keywords Facility location · Local search · Approximation algorithms · Uniform capacities

---

Work done as part of the "Approximation Algorithms" partner group of MPI-Informatik, Germany.

A. Aggarwal
Tower Research Capital LLC, Gurgaon, India

A. Louis
Georgia Tech, Atlanta, GA, USA

M. Bansal · N. Gupta
University of Delhi, Delhi, India

N. Garg (✉)
Indian Institute of Technology Delhi, Delhi, India
e-mail: naveen@cse.iitd.ac.in

S. Gupta
Google, Mountain View, CA, USA

S. Jain
Google, London, UK

Mathematics Subject Classification (2000) 90B80 · 90C59 · 68W25 · 68W40 · 68Q25

1 Introduction

In a facility location problem we are given a set of clients C and facility locations F. Opening a facility at location $i \in F$ costs f_i (the *facility cost*). The cost of servicing a client j by a facility i is given by $c_{i,j}$ (the *service cost*) and these costs form a metric i.e. for facilities i, i' and clients j, j', $c_{i',j'} \leq c_{i',j} + c_{i,j} + c_{i,j'}$. The objective is to determine which locations to open facilities in, so that the total cost for opening the facilities and for serving all the clients is minimized. Note that in this setting each client would be served by the open facility which offers the smallest service cost.

When the number of clients that a facility can serve is bounded, we have a *capacitated facility location problem*. In this paper we assume that these capacities are the same, U, for all facilities. For this problem of uniform capacities the first approximation algorithm was due to Korupolu et al. [4] who analyzed a local search algorithm and proved that any locally optimum solution has cost no more than 8 times the facility cost plus 5 times the service cost of an (global) optimum solution. In this paper we refer to such a guarantee as a (8,5)-approximation; note that this is different from the bi-criterion guarantees for which this notation is typically used. Chudak and Williamson [3] strengthened the analysis in [4] to obtain a (6,5)-approximation. Charikar and Guha [2] gave a general technique for scaling facility costs that improves the approximation guarantee to $3 + 2\sqrt{2}$.

Given the set of open facilities, the best way of serving the clients, can be determined by solving an assignment problem. Thus any solution is completely determined by the set of open facilities. The local search procedure proposed by Korupolu et al. starts with an arbitrary set of open facilities and then updates this set, using one of the operations `add`, `delete`, `swap`, whenever that operation reduces the total cost of the solution. We show that a solution which is locally optimum with respect to this same set of operations is a (3,3)-approximation. We then show that our analysis of this local search algorithm is best possible by demonstrating an instance where the locally optimum solution is 3 times the (global) optimum solution.

When facilities have different capacities, the best result known is a (6,5)-approximation by Zhang et al. [8]. The local search in this case relies on a multi-exchange operation, in which, loosely speaking, a subset of facilities from the current solution is exchanged with a subset not in the solution. This result improves on a (8,7)-approximation by Mahdian and Pal [5] and a (9,5) approximation by Pal et al. [7].

For capacitated facility location, the only algorithms known are based on local search. One version of capacitated facility location arises when we are allowed to make multiple copies of the facilities. Thus if facility i has capacity U_i and opening cost f_i, then to serve $k > U_i$ clients by facility i we need to open $\lceil k/U_i \rceil$ copies of i and incur an opening cost $f_i \lceil k/U_i \rceil$. This version is usually referred to as "facility location with soft capacities" and the best known algorithm for this problem is a 2-approximation [6].

All earlier work for capacitated facility location (uniform or non-uniform) reroutes all clients in a swap operation from the facility which is closing to one of the facilities being opened. This however can be quite expensive and cannot lead to the tight bounds that we achieve in this paper. We use the idea of Arya et al. [1] to reassign some clients of the facility being closed in a swap operation to other facilities in our current solution. However, to be able to handle the capacity constraints in this reassignment we need to extend the notion of the mapping between clients used in [1] to a fractional assignment. As in earlier work, we use the fact that when we have a local optimum, no operation leads to an improvement in cost. However, we now take carefully defined *linear combinations* of the inequalities capturing this local optimality. All previous work that we are aware of seems to only use the *sum* of such inequalities and therefore requires additional properties like the integrality of the assignment polytope to identify suitable swaps [3]. Our approach is therefore more general and amenable to better analysis. The idea of doing things fractionally appears more often in our analysis. Thus, when analyzing the cost of an operation we assign clients fractionally to the facilities and rely on the fact that such a fractional assignment cannot be better than the optimum assignment which follows from the integrality of the assignment polytope.

In Sect. 5 we give a tight example that requires the construction of a suitable set-system. While this construction itself is quite straightforward, this is the first instance we know of where such an idea has been applied to prove a large locality gap.

2 Preliminaries

Let C be the set of clients and F denote the facility locations. Let S (resp. O) be the set of open facilities in our solution (resp. optimum solution). We abuse notation and use S (resp. O) to denote our solution (resp. optimum solution). Initially S is an arbitrary set of facilities which can serve all the clients. Let cost(S) denote the total cost (facility plus service) of solution S. The three operations that make up our local search algorithm are

Add For $s \notin S$, if $\text{cost}(S + \{s\}) < \text{cost}(S)$ then $S \leftarrow S + \{s\}$.
Delete For $s \in S$, if $\text{cost}(S - \{s\}) < \text{cost}(S)$ then $S \leftarrow S - \{s\}$.
Swap For $s \in S$ and $s' \notin S$, if $\text{cost}(S - \{s\} + \{s'\}) < \text{cost}(S)$ then $S \leftarrow S - \{s\} + \{s'\}$.

S is locally optimum if none of the three operations are possible and at this point our algorithm stops.

We use $f_i, i \in F$ to denote the cost of opening a facility at location i. Let S_j, O_j denote the service-cost of client j in the solutions S and O, respectively. The presence of the `add` operation ensures that the total service cost of the clients in any locally optimum solution is at most the total cost of the optimum solution [4]. Formally,

Lemma 1 ([4]) *For any locally optimum solution S,* $\sum_{j\in C} S_j \leq \sum_{j\in C} O_j + \sum_{o\in O} f_o$.

We reprove this Lemma in Sect. 4.

Hence, most of the effort in this paper is towards bounding the facility cost of a locally optimum solution which we show is no more than 2 times the cost of an optimum solution. We prove this by identifying a suitable set of local operations and determine the increase in cost if these operations were to be performed. Since the solution is locally optimum, the increase in cost due to these operations is non-negative.[1] This gives us a set of inequalities and a suitable linear combination of these inequalities yields the bound on the facility cost of the locally optimum solution. Note that the inequalities generated are only for the purpose of analysis; we do not actually perform those local operations since we are already at a locally optimum solution.

Combining the bounds of the service cost and the facility cost of a locally optimum solution then gives us our main theorem.

Theorem 1 *For any locally optimum solution S and an optimum solution O to the facility location problem with uniform capacities, cost(S) ≤ 3cost(O).*

To ensure that our procedure has a polynomial running time we use an idea first proposed in [4]—a local step is performed only if the cost of the solution reduces by more than $(\epsilon/4n)\text{cost}(S)$ where $\epsilon > 0$ and $n = |F|$ is the number of facility locations. It is immediate that as a result of this modification the number of local search steps done is at most $4n\epsilon^{-1}\log(\text{cost}(S_0)/\text{cost}(O))$ where S_0 is the initial solution. In Sect. 3 we argue the approximation guarantee of this modified local search procedure increases to at most $3/(1-\epsilon)$.

The rest of the paper is organized as follows. In Sect. 3 we bound the facility costs of the locally optimum solution assuming that the facilities in the locally optimum solution, S are disjoint from the facilities of the optimum solution, O. Most of the new ideas in the paper appear in this section. In Sect. 4 we extend the argument to the case when the facilities in S and O are not disjoint. In Sect. 5 we give an example of a solution which is locally optimal with respect to the operations of `add, delete, swap` and has cost three times the optimum. This establishes that our analysis is tight.

3 Bounding the facility costs

Let S denote the locally optimum solution obtained. For the rest of this section we assume that the sets S and O are disjoint. This assumption allows us to add any facility of O or to swap any facility in S with a facility in O without worrying about the possibility that the facility of O included in our solution might already be part of S.

Let $N_S(s)$ denote the clients served by facility s in the solution S and $N_O(o)$ denote the clients served by facility o in solution O. Let N_s^o denote the set of clients served by facility s in solution S and by facility o in solution O. We will associate a weight, $\texttt{wt} : C \to [0..1]$, with each client which satisfies the following properties.

1. For a client $j \in C$ let $\sigma(j)$ be the facility which serves j in solution S. Then

$$\texttt{wt}(j) \leq \min\left(1, \frac{U - |N_S(\sigma(j))|}{|N_S(\sigma(j))|}\right).$$

[1] In fact, we do not determine the exact increase in cost when a local operation is performed but only an upperbound on this quantity.

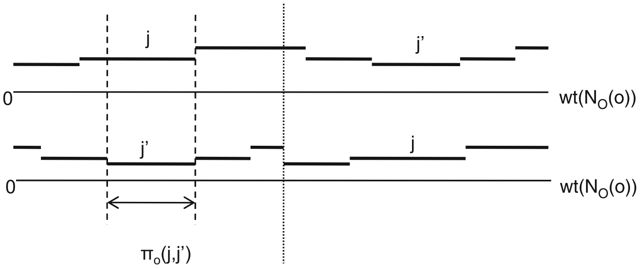

Fig. 1 Defining π_O. The lower arrangement is obtained by splitting the top arrangement at the central *dotted line* and swapping the two halves

Let $\texttt{init-wt}(j)$ denote the quantity on the right of the above inequality. Since $|N_S(\sigma(j))| \leq U$, we have that $0 \leq \texttt{init-wt}(j) \leq 1$.

2. For all $o \in O$ and $s \in S$, $\texttt{wt}(N_s^o) \leq \texttt{wt}(N_O(o))/2$. Here for $X \subseteq C$, $\texttt{wt}(X)$ denotes the sum of the weights of the clients in X.

To determine $\texttt{wt}(j)$ so that these two properties are satisfied we start by assigning $\texttt{wt}(j) = \texttt{init-wt}(j)$. However, this assignment might violate the second property. A facility $s \in S$ *captures* a facility $o \in O$ if $\texttt{init-wt}(N_s^o) > \texttt{init-wt}(N_O(o))/2$. Note that at most one facility in S can capture a facility o. If s does not capture o then for all $j \in N_s^o$ define $\texttt{wt}(j) = \texttt{init-wt}(j)$. However if s captures o then for all $j \in N_s^o$ define $\texttt{wt}(j) = \alpha \cdot \texttt{init-wt}(j)$ where $\alpha < 1$ is such that $\texttt{wt}(N_s^o) = \texttt{wt}(N_O(o))/2$. Note that if $N_s^o = N_O(o)$ then $\alpha = 0$.

For a facility $o \in O$ we define a fractional assignment $\pi_o : N_O(o) \times N_O(o) \to \Re^+$ with the following properties.

separation $\pi_o(j, j') > 0$ only if j and j' are served by different facilities in S.

balance $\sum_{j' \in N_O(o)} \pi_o(j', j) = \sum_{j' \in N_O(o)} \pi_o(j, j') = \texttt{wt}(j)$ for all $j \in N_O(o)$.

The fractional assignment π_o can be obtained along the same lines as the mapping in [1]. Associate an interval of length $\texttt{wt}(j)$ for each $j \in N_O(o)$ and arrange these intervals on a line segment of length $\texttt{wt}(N_O(o))$ (see Fig. 1). The intervals are ordered so that intervals corresponding to clients served by the same facility in S appear together. Consider another arrangement of intervals obtained from the first by splitting the line segment at the center and swapping the two halves. As a consequence, one interval might be split and be non-contiguous in the second arrangement. Superimpose these two arrangements. $\pi_o(j, j')$ is now defined as the overlap between the interval corresponding to j in the first arrangement and the interval j' in the second. The second property of the weights ensures that there is no overlap between an interval in the first arrangement and the corresponding interval in the second arrangement. Further, it is easy to see that the mapping π_o as defined here satisfies the properties of separation and balance.

The individual fractional assignments π_o are extended to a fractional assignment over all clients, $\pi : C \times C \to \Re^+$ in the obvious way—$\pi(j, j') = \pi_o(j, j')$ if $j, j' \in N_O(o)$ and is 0 otherwise.

To bound the facility cost of a facility $s \in S$ we will close the facility and assign the clients served by s to other facilities in S and, maybe, some facility in O. The reassignment of the clients served by s to the facilities in S is done using the fractional assignment π. Thus if client j is served by s in the solution S and $\pi(j, j') > 0$ then we assign a $\pi(j, j')$ fraction of j to the facility $\sigma(j')$. Note that

1. $\sigma(j') \neq s$ and this follows from the separation property of π.
2. j is reassigned to the facilities in S to a total extent of $\texttt{wt}(j)$ (balance property).
3. A facility $s' \in S, s' \neq s$, would get some additional clients. The total extent to which these additional clients are assigned to s' is at most $\texttt{wt}(N_S(s'))$ (balance property). Since

$$\texttt{wt}(N_S(s')) \leq \texttt{init-wt}(N_S(s')) \leq U - \left|N_S(s')\right|,$$

the total number of clients assigned to s' after this reassignment is at most U.

Let $\Delta(s)$ denote the increase in the service-cost of the clients served by s due to the above reassignment.

Lemma 2 $\sum_{s\in S} \Delta(s) \leq \sum_{j\in C} 2O_j\texttt{wt}(j)$

Proof Let $\pi(j, j') > 0$. When the facility $\sigma(j)$ is closed and $\pi(j, j')$ fraction of client j assigned to facility $\sigma(j')$, the increase in service cost is $\pi(j, j')(c_{j,\sigma(j')} - c_{j,\sigma(j)})$. Since $c_{j,\sigma(j')} \leq O_j + O_{j'} + S_{j'}$ we have

$$\begin{aligned}\sum_{s\in S} \Delta(s) &= \sum_{j,j'\in C} \pi(j, j')(c_{j,\sigma(j')} - c_{j,\sigma(j)})\\ &\leq \sum_{j,j'\in C} \pi(j, j')(O_j + O_{j'} + S_{j'} - S_j)\\ &= 2\sum_{j\in C} O_j\texttt{wt}(j)\end{aligned}$$

where the last equality follows from the balance property. □

If $\texttt{wt}(j) < 1$ then some part of j remains unassigned. The quantity $1 - \texttt{wt}(j)$ is the *residual weight* of client j and is denoted by $\texttt{res-wt}(j)$. Clearly $0 \leq \texttt{res-wt}(j) \leq 1$. Note that

1. If we close facility $s \in S$ and assign the residual weight of all clients served by s to a facility $o \in O$ then the total extent to which clients are assigned to o equals $\texttt{res-wt}(N_S(s))$ which is less than U.
2. Define

$$c_{s,o} = \min_{j\in C}(c_{j,s} + c_{j,o}).$$

The service cost of a client j, which is assigned to o instead of s would increase by $c_{j,o} - c_{j,s}$. Since service costs satisfy the metric property, for all clients j,

$$c_{j,o} - c_{j,s} \leq c_{s,o}.$$

3. The total increase in service cost of all clients in $N_S(s)$ which are assigned (partly) to o is at most $c_{s,o}\texttt{res-wt}(N_S(s))$.

Let $\langle s, o\rangle$ denote the swapping of facilities s, o and the reassignment of clients served by s to facilities in $S-\{s\}\cup\{o\}$ as discussed above. Since S is locally optimum we have

$$f_o - f_s + c_{s,o}\texttt{res-wt}(N_S(s)) + \Delta(s) \geq 0. \tag{1}$$

The above inequalities are written for every pair (s, o), $s \in S$, $o \in O$. We take a linear combination of these inequalities with the inequality corresponding to $\langle s, o\rangle$ having a weight $\lambda_{s,o}$ in the combination to get

$$\sum_{s,o}\lambda_{s,o}f_o - \sum_{s,o}\lambda_{s,o}f_s + \sum_{s,o}\lambda_{s,o}c_{s,o}\texttt{res-wt}(N_S(s)) + \sum_{s,o}\lambda_{s,o}\Delta(s) \geq 0. \tag{2}$$

where

$$\lambda_{s,o} = \frac{\texttt{res-wt}(N_s^o)}{\texttt{res-wt}(N_S(s))}$$

and is 0 if $\texttt{res-wt}(N_S(s)) = 0$. Let S' be the subset of facilities in the solution S for which $\texttt{res-wt}(N_S(s)) = 0$. A facility $s \in S'$ can be deleted from S and its clients reassigned completely to the other facilities in S. This implies

$$-f_s + \Delta(s) \geq 0 \tag{3}$$

We write such an inequality for each $s \in S'$ and add them to inequality (2).

Note that for all $s \in S - S'$, $\sum_o \lambda_{s,o} = 1$. This implies that

$$\sum_{s\in S'} f_s + \sum_{s,o}\lambda_{s,o}f_s = \sum_s f_s \tag{4}$$

and

$$\sum_{s\in S'}\Delta(s) + \sum_{s,o}\lambda_{s,o}\Delta(s) = \sum_s \Delta(s) \leq \sum_{j\in C} 2O_j\texttt{wt}(j) \tag{5}$$

However, the reason for defining $\lambda_{s,o}$ as above is to ensure the following property.

Lemma 3 $\sum_{s,o}\lambda_{s,o}c_{s,o}\texttt{res-wt}(N_S(s)) \leq \sum_{j\in C}\texttt{res-wt}(j)(O_j + S_j)$

Proof The left hand side in the inequality is $\sum_{s,o} c_{s,o}\texttt{res-wt}(N_s^o)$. Since for each client $j \in N_s^o$, $c_{s,o} \leq O_j + S_j$ we have

$$\begin{aligned} c_{s,o}\texttt{res-wt}(N_s^o) &= \sum_{j\in N_s^o} c_{s,o}\texttt{res-wt}(j) \\ &\leq \sum_{j\in N_s^o}\texttt{res-wt}(j)(O_j + S_j) \end{aligned}$$

which, when summed over all s and o implies the Lemma. □

Incorporating equations (4), (5) and Lemma 3 into inequality (2) we get

$$\begin{aligned}\sum_s f_s &\le \sum_{s,o} \lambda_{s,o} f_o + \sum_{j\in C} \texttt{res-wt}(j)(O_j + S_j) + \sum_{j\in C} 2O_j \texttt{wt}(j)\\ &= \sum_{s,o} \lambda_{s,o} f_o + 2\sum_{j\in C} O_j + \sum_{j\in C} \texttt{res-wt}(j)(S_j - O_j) \end{aligned} \tag{6}$$

We now need to bound the number of times a facility of the optimum solution may be opened.

Lemma 4 *For all $o \in O$, $\sum_s \lambda_{s,o} \le 2$.*

Proof We begin with the following observations.

1. For all s, o, $\lambda_{s,o} \le 1$.
2. Let $I \subseteq S$ be the facilities s such that s does not capture o and $|N_S(s)| \le U/2$. Let $s \in I$ and $j \in N_s^o$. Note that $\texttt{wt}(j) = \texttt{init-wt}(j) = 1$ and so $\texttt{res-wt}(j) = 0$. This implies that $\texttt{res-wt}(N_s^o) = 0$ and so for all $s \in I$, $\lambda_{s,o} = 0$.

Thus we only need to show that $\sum_{s\notin I} \lambda_{s,o} \le 2$. We now consider two cases.

1. o is not captured by any $s \in S$. Let s be a facility not in I which does not capture o. For $j \in N_s^o$,

$$\texttt{res-wt}(j) = 1 - \texttt{wt}(j) = 1 - \texttt{init-wt}(j) = 2 - \frac{U}{|N_S(s)|}.$$

However, for $j \in N_S(s)$ we have that

$$\texttt{res-wt}(j) = 1 - \texttt{wt}(j) \ge 1 - \texttt{init-wt}(j) = 2 - \frac{U}{|N_S(s)|}.$$

Therefore

$$\lambda_{s,o} \le \frac{|N_s^o|}{|N_S(s)|}$$

Hence

$$\sum_s \lambda_{s,o} = \sum_{s\notin I} \lambda_{s,o} \le \sum_{s\notin I} \frac{|N_s^o|}{|N_S(s)|} \le \sum_{s\notin I} \frac{|N_s^o|}{U/2} \le \frac{|N_O(o)|}{U/2} \le 2.$$

2. o is captured by $s' \in S$. This implies

$$\begin{aligned}\texttt{init-wt}(N_{s'}^o) &\ge \sum_{s\ne s'} \texttt{init-wt}(N_s^o)\\ &\ge \sum_{s\notin I\cup\{s'\}} \texttt{init-wt}(N_s^o)\end{aligned}$$

$$= \sum_{s \notin I \cup \{s'\}} \left|N_s^o\right| \frac{U - |N_S(s)|}{|N_S(s)|}$$

$$= \sum_{s \notin I \cup \{s'\}} \left(U \frac{\left|N_s^o\right|}{|N_S(s)|} - \left|N_s^o\right| \right)$$

Since $\texttt{init-wt}(N_{s'}^o) \leq \left|N_{s'}^o\right|$ rearranging we get,

$$\sum_{s \notin I \cup \{s'\}} \frac{\left|N_s^o\right|}{|N_S(s)|} \leq \sum_{s \notin I} \frac{\left|N_s^o\right|}{U} \leq 1.$$

Now

$$\sum_{s \notin I \cup \{s'\}} \lambda_{s,o} \leq \sum_{s \notin I \cup \{s'\}} \frac{\left|N_s^o\right|}{|N_S(s)|} \leq 1$$

and since $\lambda_{s',o} \leq 1$ we have

$$\sum_s \lambda_{s,o} = \sum_{s \notin I} \lambda_{s,o} \leq 2.$$

This completes the proof. □

Incorporating Lemma 4 into inequality (6) we get

$$\sum_s f_s \leq 2 \left(\sum_o f_o + \sum_{j \in C} O_j \right) + \sum_{j \in C} \texttt{res-wt}(j)(S_j - O_j) \tag{7}$$

Note that $\sum_{j \in C} \texttt{res-wt}(j)(S_j - O_j)$ is at most $\sum_{j \in C}(S_j - O_j)$ which in turn can be bounded by $\sum_o f_o$ by considering the operation of adding facilities in the optimum solution. This, however, would lead to a bound of $3 \sum_o f_o + 2 \sum_{j \in C} O_j$ on the facility cost of our solution.

The key to obtaining a sharper bound on the facility cost of our solution is the observation that in the swap $\langle s, o \rangle$ facility o gets only $\texttt{res-wt}(N_S(s))$ clients and so can accommodate an additional $U - \texttt{res-wt}(N_S(s))$ clients. Since we need to bound $\sum_{j \in C} \texttt{res-wt}(j)(S_j - O_j)$, we assign the clients in $N_O(o)$ to facility o in the ratio of their residual weights. Thus client j would be assigned to an extent $\beta_{s,o}\texttt{res-wt}(j)$ where

$$\beta_{s,o} = \min \left(1, \frac{U - \texttt{res-wt}(N_S(s))}{\texttt{res-wt}(N_O(o))} \right).$$

$\beta_{s,o}$ is defined so that o gets at most U clients. Let $\Delta'(s, o)$ denote the increase in service cost of the clients of $N_O(o)$ due to this reassignment. Hence

$$\Delta'(s,o) = \beta_{s,o} \sum_{j \in N_O(o)} \texttt{res-wt}(j)(O_j - S_j). \tag{8}$$

The inequality (1) corresponding to the swap $\langle s, o\rangle$ would now get an additional term $\Delta'(s,o)$ on the left. Hence the term $\sum_{s,o} \lambda_{s,o}\Delta'(s,o)$ would appear on the left in inequality (2) and on the right in inequality (6).

Now

$$\begin{aligned}\sum_s \lambda_{s,o}\Delta'(s,o) &= \sum_s \left(\lambda_{s,o}\beta_{s,o} \sum_{j\in N_O(o)} \texttt{res-wt}(j)(O_j - S_j)\right)\\ &= \left(\sum_s \lambda_{s,o}\beta_{s,o}\right) \sum_{j\in N_O(o)} \texttt{res-wt}(j)(O_j - S_j).\end{aligned}$$

If $\sum_s \lambda_{s,o}\beta_{s,o} > 1$ then we reduce some $\beta_{s,o}$ so that the sum is exactly 1 (we will later show that this does not affect the analysis). On the other hand if $\sum_s \lambda_{s,o}\beta_{s,o} = 1-\gamma_o$, $\gamma_o > 0$, then we take the inequalities corresponding to the operation of adding the facility $o \in O$

$$f_o + \sum_{j\in N_O(o)} \texttt{res-wt}(j)(O_j - S_j) \geq 0 \tag{9}$$

and add these to inequality (2) with a weight γ_o. Hence the total increase in the left hand side of inequality (2) is

$$\begin{aligned}&\sum_{s,o} \lambda_{s,o}\Delta'(s,o) + \sum_o \gamma_o \left(f_o + \sum_{j\in N_O(o)} \texttt{res-wt}(j)(O_j - S_j)\right)\\ &\quad= \sum_o \sum_{j\in N_O(o)} (1-\gamma_o)\texttt{res-wt}(j)(O_j - S_j)\\ &\qquad+ \sum_o \gamma_o f_o + \sum_o \sum_{j\in N_O(o)} \gamma_o \texttt{res-wt}(j)(O_j - S_j)\\ &\quad= \sum_o \sum_{j\in N_O(o)} \texttt{res-wt}(j)(O_j - S_j) + \sum_o \gamma_o f_o\\ &\quad= \sum_{j\in C} \texttt{res-wt}(j)(O_j - S_j) + \sum_o \gamma_o f_o\end{aligned}$$

and so inequality (6) now becomes

$$\begin{aligned}\sum_s f_s \leq& \sum_o \sum_s \lambda_{s,o} f_o + 2\sum_{j\in C} O_j + \sum_o \gamma_o f_o\\ &+ \sum_{j\in C} \texttt{res-wt}(j)(S_j - O_j) + \sum_{j\in C} \texttt{res-wt}(j)(O_j - S_j)\end{aligned}$$

$$
\begin{aligned}
&= \sum_o \left(\gamma_o + \sum_s \lambda_{s,o}\right) f_o + 2\sum_{j\in C} O_j \\
&= \sum_o \left(1 + \sum_s \lambda_{s,o}(1-\beta_{s,o})\right) f_o + 2\sum_{j\in C} O_j \\
&\le 2\left(\sum_o f_o + \sum_{j\in C} O_j\right)
\end{aligned}
$$

where the last inequality follows from the following Lemma.

Lemma 5 $\sum_s \lambda_{s,o}(1-\beta_{s,o}) \le 1$.

Proof When $\sum_s \lambda_{s,o}\beta_{s,o} > 1$ we reduced some $\beta_{s,o}$ to ensure that the sum is exactly 1. In this case

$$\sum_s \lambda_{s,o}(1-\beta_{s,o}) = \sum_s \lambda_{s,o} - 1 \le 1,$$

since by Lemma 4, $\sum_s \lambda_{s,o} \le 2$.

We now assume that no $\beta_{s,o}$ was reduced. Since

$$\texttt{res-wt}(N_O(o)) \le |N_O(o)| \le U$$

we have

$$
\begin{aligned}
\beta_{s,o} &= \min\left(1, \frac{U - \texttt{res-wt}(N_S(s))}{\texttt{res-wt}(N_O(o))}\right) \\
&\ge \min\left(1, 1 - \frac{\texttt{res-wt}(N_S(s))}{\texttt{res-wt}(N_O(o))}\right) \\
&= 1 - \frac{\texttt{res-wt}(N_S(s))}{\texttt{res-wt}(N_O(o))}
\end{aligned}
$$

Hence

$$\sum_s \lambda_{s,o}(1-\beta_{s,o}) \le \sum_s \frac{\texttt{res-wt}(N_s^o)}{\texttt{res-wt}(N_O(o))} = 1.$$

□

This completes the proof of the following theorem.

Theorem 2 *When $S \cap O = \phi$, the total cost of open facilities in any locally optimum solution is at most twice the cost of an optimum solution.*

Recall that to ensure that the local search procedure has a polynomial running time we modified the local search procedure so that a step was performed only when the cost of the solution decreases by at least $(\epsilon/4n)\text{cost}(S)$. This modification implies that the right hand sides of inequalities (1), (3) and (9) which are all zero should instead be $(-\epsilon/4n)\text{cost}(S)$. Note that for every choice of $s \in S$ and $o \in O$ we add a $\lambda_{s,o}$ multiple of inequality (1) to obtain inequality (2). Since $\sum_o \lambda_{s,o} = 1$, hence $\sum_{o,s} \lambda_{s,o} = |S| \leq n$. We also add inequality (3) for every $s \in S$ to inequality (2). Similarly, for every $o \in O$, a γ_o ($\gamma_o \leq 1$) multiple of inequality (9) is added to inequality (2).

Putting all these modifications together gives rise to an extra term of at most $(3\epsilon/4)\text{cost}(S)$. This implies that the facility cost of solution S is at most $2\text{cost}(O) + (3\epsilon/4)\text{cost}(S)$. Similarly, the service cost of solution S can now be bounded by $\text{cost}(O) + (\epsilon/4)\text{cost}(S)$. Adding these yields

$$(1-\epsilon)\text{cost}(S) \leq 3\text{cost}(O)$$

which implies that S is a $3/(1-\epsilon)$ approximation to the optimum solution.

4 When $S \cap O \neq \phi$

We now consider the case when $S \cap O \neq \phi$. We construct a bipartite graph, G, on the vertex set $C \cup F$ as in [3]. Every client $j \in C$ has an edge from the facility $\sigma(j) \in S$ and an edge to the facility $\tau(j) \in O$, where $\tau(j)$ is the facility in O serving client j. Thus each client has one incoming and one outgoing edge. A facility $s \in S$ has $|N_S(s)|$ outgoing edges and a facility $o \in O$ has $|N_O(o)|$ incoming edges. Decompose the edges of G into a set of maximal paths, $\mathcal{P}$, and cycles, $\mathcal{C}$. Note that all facilities on a cycle are from $S \cap O$. Consider a maximal path, $p \in \mathcal{P}$ which starts at a vertex $s \in S$ and ends at a vertex $o \in O$. Let $\text{head}(p)$ denote the client served by s on this path and $\text{tail}(p)$ be the client served by o on this path. Let $s_0, j_0, s_1, j_1, \ldots, s_k, j_k, o$ be the sequence of vertices on this path where $s = s_0$. Note that $\{s_1, s_2, \ldots, s_k\} \subseteq S \cap O$. A *shift* along this path is a reassignment of clients so that j_i which was earlier assigned to s_i is now assigned to s_{i+1} where $s_{k+1} = o$. As a consequence of this shift, facility s serves one less client while facility o serves one more client. Let $\text{shift}(p)$ denote the increase in service cost due to a shift along the path p. Then

$$\text{shift}(p) = \sum_{c \in C \cap p} (O_c - S_c).$$

We can similarly define a shift along a cycle. The increase in service cost equals the sum of $O_j - S_j$ for all clients j in the cycle and since the assignment of clients to facilities is done optimally in our solution and in the global optimum this sum is zero. Thus

$$\sum_{j \in \mathcal{C}} (O_j - S_j) = 0. \tag{10}$$

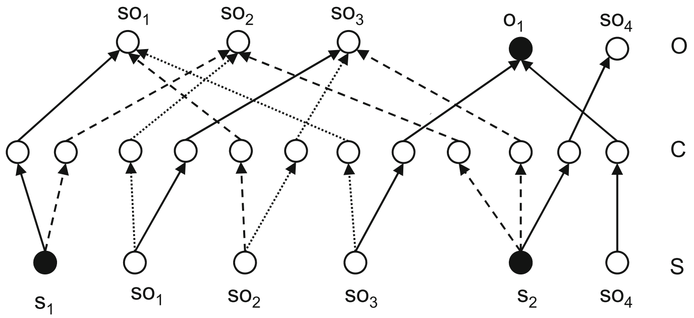

Fig. 2 An instance showing the decomposition into cycles (*dotted arcs*), swap paths (*solid arcs*) and transfer paths (*dashed arcs*). The facilities labeled so_1, so_2, so_3 and so_4 are in $S \cap O$ and have been duplicated. The cycle is so_1, so_2, so_3, so_1. The transfer paths are (s_1, so_2, so_1), (s_2, so_2) and (s_2, so_3). The swap paths are s_1, so_1, so_3, o_1 and s_2, so_4, o_1

Consider the operation of adding a facility $o \in O$. We shift along all paths which end at o. The increase in service cost due to these shifts equals the sum of $O_j - S_j$ for all clients j on these paths and this quantity is at least $-f_o$.

$$\sum_{j \in \mathcal{P}} (O_j - S_j) \geq -\sum_{o \in O} f_o. \tag{11}$$

Thus

$$\sum_{j \in C} (O_j - S_j) = \sum_{j \in \mathcal{P}} (O_j - S_j) + \sum_{j \in \mathcal{C}} (O_j - S_j) \geq -\sum_{o \in O} f_o$$

which implies that the service cost of S is bounded by $\sum_{o \in O} f_o + \sum_{j \in C} O_j$.

To bound the cost of facilities in $S - O$ we only need the paths that start from a facility in $S - O$. Hence we throw away all cycles and all paths that start at a facility in $S \cap O$; this is done by removing all clients on these cycles and paths. Let $\mathcal{P}$ denote the remaining paths and C the remaining clients. Every client in C either belongs to a path which ends in $S \cap O$ (*transfer* path) or to a path which ends in $O - S$ (*swap* path). Let $\mathcal{T}$ denote the set of transfer paths and $\mathcal{S}$ the set of swap paths (see Fig. 2).

We now define N_s^o to be the set of paths that start at $s \in S$ and end at $o \in O$. Further, define

$$N_S(s) = \cup_{o \in O - S} N_s^o.$$

Note that we do not include the transfer paths in the above definition. Similarly for all $o \in O$ define

$$N_O(o) = \cup_{s \in S - O} N_s^o.$$

Just as we defined the `init-wt()`, `wt()` and `res-wt()` of a client, we can define the `init-wt()`, `wt()` and `res-wt()` of a swap path. Thus for a path p which starts from $s \in S - O$ we define

$$\texttt{init-wt}(p) = \min\left(1, \frac{U - |N_S(s)|}{|N_S(s)|}\right).$$

The notion of *capture* remains the same and we reduce the initial weights on the paths to obtain their weights. Thus $\texttt{wt}(p) \leq \texttt{init-wt}(p)$ and for every $s \in S$ and $o \in O$, $\texttt{wt}(N_s^o) \leq \texttt{wt}(N_O(o))/2$. For every $o \in O - S$ we define a fractional mapping $\pi_o : N_O(o) \times N_O(o) \to \Re^+$ such that

separation $\pi_o(p, p') > 0$ only if p and p' start at different facilities in $S - O$.
balance $\sum_{p' \in N_O(o)} \pi_o(p', p) = \sum_{p' \in N_O(o)} \pi_o(p, p') = \texttt{wt}(p)$ for all $p \in N_O(o)$.

This fractional mapping can be constructed in the same way as done earlier. The way we use this fractional mapping, π, will differ slightly. When facility s is closed, we will use π to partly reassign the clients served by s in the solution S to other facilities in S. If p is a path starting from s and $\pi(p, p') > 0$, then we shift along p and the client tail(p) is assigned to s', where s' is the facility from which p' starts. This whole operation is done to an extent of $\pi(p, p')$. The cost of assigning client tail(p) to s' can be bounded by the sum of the service cost of tail(p) in solution O and the *length* of the path p' where

$$\text{length}(p') = \sum_{c \in C \cap p'} (O_c + S_c).$$

Let $\Delta(s)$ denote the total increase in service cost due to the reassignment of clients on all swap paths starting from s. Then

$$\begin{aligned}\sum_s \Delta(s) &\leq \sum_s \sum_{p \in N_S(s)} \sum_{p' \in \mathcal{P}} \pi(p, p')(\text{shift}(p) + \text{length}(p')) \\ &= \sum_{p \in \mathcal{S}} \texttt{wt}(p)\,(\text{shift}(p) + \text{length}(p)) \qquad (12)\end{aligned}$$

As a result of the above reassignment a facility $s' \in S - O, s' \neq s$ might get additional clients whose "number" is at most $\texttt{wt}(N_S(s'))$. Note that this is less than $\texttt{init-wt}(N_S(s'))$ which is at most $U - \left|N_S(s')\right|$. The number of clients s' was serving equals $\left|N_S(s')\right| + \left|T(s')\right|$ where $T(s')$ is the set of transfer paths starting from s'. This implies that the total number of clients s' would have after the reassignment could exceed U. To prevent this violation of our capacity constraint, we also perform a shift along these transfer paths (Fig. 3).

Suppose s' gets an additional client, say tail(p), to an extent of $\pi(p, p')$, where $p' \in N_S(s')$. Then for all paths $q \in T(s')$, we would shift along path q to an extent $\pi(p, p')/\texttt{wt}(N_S(s'))$. This ensures that

1. The total extent to which we will shift along a path $q \in T(s')$ is given by

$$\sum_p \sum_{p' \in N_S(s')} \frac{\pi(p, p')}{\texttt{wt}(N_S(s'))}$$

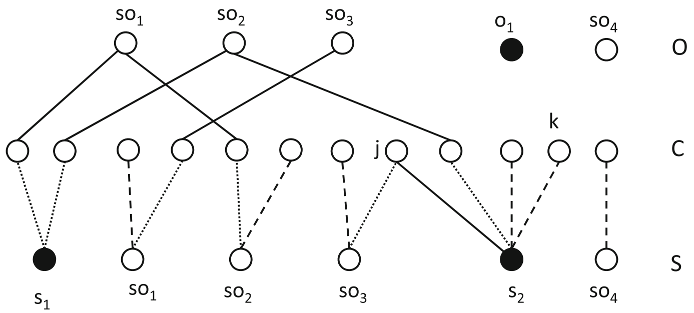

Fig. 3 The figure shows the assignment of clients to facilities after facility s_1 has been closed ($U = 3$) in the instance given in Fig. 2. The *dotted lines* show the earlier assignment while the *solid lines* show the new assignment. Those assignments which do not change are shown with *dashed lines*. Note that o serves two clients j, k which are the heads of swap paths s_1, so_1, so_3, o_1 and s_2, so_4, o_1. These two clients are mapped to each other in the mapping π. When facility s_1 is closed we perform a shift along transfer path s_1, so_2, so_1. We also perform a shift along the swap path s_1, so_1, so_3, o_1 with the last client on this path, j now assigned to s_2. Since, s_2 was already serving 3 clients, we move one of its clients along a transfer path (s_2, so_2)

which is at most 1. This in turn implies that we do not violate the capacity of any facility in $S \cap O$. This is because, if there are t transfer paths ending at a facility $o \in S \cap O$ then o serves t more clients in solution O than in S. Hence, in solution S, o serves at most $U - t$ clients. Since the total extent to which we could shift along a transfer path ending at o is 1, even if we were to perform shift along all transfer paths ending in o, the capacity of o in our solution S would not be violated.

2. The capacity constraint of no facility in $S - O$ is violated. If a facility $s' \in S - O$ gets an additional x clients as a result of reassigning the clients of some facility $s \neq s'$, then it would also lose some clients, say y, due to the shifts along the transfer paths. Now

$$y = \left|T(s')\right| \sum_{p} \sum_{p' \in N_S(s')} \frac{\pi(p, p')}{\mathtt{wt}\,(N_S(s'))} = \frac{x\left|T(s')\right|}{\mathtt{wt}\,(N_S(s'))}$$

and hence the additional number of clients served by s' is $x - y$ which equals

$$x\left(1 - \frac{\left|T(s')\right|}{\mathtt{wt}\,(N_S(s'))}\right) \leq \mathtt{wt}\,(N_S(s')) - \left|T(s')\right| \leq U - \left|N_S(s')\right| - \left|T(s')\right|,$$

where the first inequality follows from the fact that $x \leq \mathtt{wt}\,(N_S(s'))$ and the second inequality by definition of $\mathtt{wt}$. Since, initially, s' was serving $\left|N_S(s')\right| + \left|T(s')\right|$ clients, the total number of clients that s' is serving after the reassignment is at most U.

Note that when we close facility s we shift on transfer paths starting from s as well as on some transfer paths starting at $s' \neq s$. Let $\Gamma(s)$ denote the total increase in service cost due to shifts on all transfer paths when facility s is closed. Consider a transfer path, q, starting from s. We would shift once along path q when we close facility s. We would also be shifting along q to an extent of $\sum_p \sum_{p' \in N_S(s)} \pi(p, p')/\mathtt{wt}(N_S(s))$ (which is at most 1) when facilities other than s are closed. Hence,

$$\sum_s \Gamma(s) \leq 2 \sum_{q \in \mathcal{T}} \text{shift}(q) \tag{13}$$

For a swap path p, define $\mathtt{res\text{-}wt}(p) = 1 - \mathtt{wt}(p)$. Let j be head(p) and define $\mathtt{wt}(j) = \mathtt{wt}(p)$ and $\mathtt{res\text{-}wt}(j) = \mathtt{res\text{-}wt}(p)$. Let p start from facility s. When s is closed, client j is assigned to an extent $\mathtt{wt}(j)$ to other facilities in S. We will be assigning the remaining part of j to a facility $o \in O - S$ that will be opened when s is closed. Hence the total number of clients that will be assigned to o is $\mathtt{res\text{-}wt}(N_S(s))$ which is less than U. The increase in service cost due to this reassignment is at most $c_{s,o}\mathtt{res\text{-}wt}(N_S(s))$. As done earlier, the inequality corresponding to the swap $\langle s, o\rangle$ is counted to an extent $\lambda_{s,o}$ in the linear combination. Since $c_{s,o} \leq \text{length}(p)$ for all $p \in N_s^o$, we have the following equivalent of Lemma 3

$$\sum_{s,o} \lambda_{s,o} c_{s,o} \mathtt{res\text{-}wt}(N_S(s)) \leq \sum_{p \in \mathcal{S}} \mathtt{res\text{-}wt}(p)\, \text{length}(p). \tag{14}$$

The remaining available capacity of o is utilized by assigning each client $j \in N_O(o)$ to an extent $\beta_{s,o}\mathtt{res\text{-}wt}(j)$, where $\beta_{s,o}$ is defined as before. This assignment is actually done by shifting along each path, $p \in N_O(o)$, by an extent $\beta_{s,o}\mathtt{res\text{-}wt}(p)$. Let $\Delta'(s, o)$ be the increase in cost due to this reassignment of clients in $N_O(o)$. Then

$$\Delta'(s, o) \leq \beta_{s,o} \sum_{p \in N_O(o)} \mathtt{res\text{-}wt}(p)\, \text{shift}(p).$$

This operation is a part of $\langle s, o\rangle$ and hence is counted to an extent $\lambda_{s,o}$ in the linear combination. Therefore the contribution of this term is

$$\sum_{s,o} \lambda_{s,o} \Delta'(s, o) \leq \sum_o \left(\sum_s \lambda_{s,o} \beta_{s,o} \right) \sum_{p \in N_O(o)} \mathtt{res\text{-}wt}(p)\, \text{shift}(p). \tag{15}$$

Adding facility $o \in O - S$ and shifting each path $p \in N_O(o)$ by an extent $\mathtt{res\text{-}wt}(p)$ gives us the following inequality.

$$f_o + \sum_{p \in N_O(o)} \mathtt{res\text{-}wt}(p)\, \text{shift}(p) \geq 0 \tag{16}$$

As before, if $\sum_s \lambda_{s,o}\beta_{s,o} > 1$ then we reduce some $\beta_{s,o}$ so that the sum is exactly 1. Else, we add a $1 - \sum_s \lambda_{s,o}\beta_{s,o}$ multiple of inequality (16) to inequality (15) to get

$$\sum_{s,o} \lambda_{s,o} \Delta'(s,o) \le \sum_{o} \gamma_o f_o + \sum_{o} \sum_{p \in N_O(o)} \texttt{res-wt}\,(p)\,\text{shift}(p). \tag{17}$$

where $\gamma_o = \max\left\{0, 1 - \sum_s \lambda_{s,o}\beta_{s,o}\right\}$.

The inequality corresponding to the swap $\langle s, o\rangle$ is

$$f_o - f_s + c_{s,o}\texttt{res-wt}\,(N_S(s)) + \Delta(s) + \Gamma(s) + \Delta'(s,o) \ge 0,$$

and taking a linear combination of the inequalities corresponding to the swaps $\langle s, o\rangle$, $s \in S - O$, $o \in O - S$ with weights $\lambda_{s,o}$ yields

$$\sum_{s,o} \lambda_{s,o} f_o - \sum_{s,o} \lambda_{s,o} f_s + \sum_{s,o} \lambda_{s,o} c_{s,o}\texttt{res-wt}\,(N_S(s)) + \sum_{s,o} \lambda_{s,o}(\Delta(s) + \Gamma(s)) + \sum_{s,o} \lambda_{s,o}\Delta'(s,o) \ge 0.$$

Since, for all s, $\sum_o \lambda_{s,o} = 1$, we get

$$\sum_{s \in S-O} f_s \le \sum_{s,o} \lambda_{s,o} f_o + \sum_{s,o} \lambda_{s,o}\Delta'(s,o) + \sum_{s,o} \lambda_{s,o} c_{s,o}\texttt{res-wt}\,(N_S(s)) + \sum_{s}(\Delta(s) + \Gamma(s)) \tag{18}$$

Putting the bounds from inequalities (12),(13),(14) and (17) into the right hand side of inequality (18), yields

$$\begin{aligned}
\sum_{s \in S-O} f_s \le& \sum_{o \in O-S} \left(\gamma_o + \sum_s \lambda_{s,o}\right) f_o + \sum_{p \in \mathcal{S}} \texttt{res-wt}\,(p)\,\text{shift}(p) \\
&+ \sum_{p \in \mathcal{S}} \texttt{res-wt}\,(p)\,\text{length}(p) + \sum_{p \in \mathcal{S}} \texttt{wt}\,(p)\,(\text{shift}(p) + \text{length}(p)) \\
&+ 2\sum_{p \in \mathcal{T}} \text{shift}(p) \\
\le& \; 2\sum_{o \in O-S} f_o + \sum_{p \in \mathcal{S}} \texttt{res-wt}\,(p)\,(\text{shift}(p) + \text{length}(p)) \\
&+ \sum_{p \in \mathcal{S}} \texttt{wt}\,(p)\,(\text{shift}(p) + \text{length}(p)) + 2\sum_{p \in \mathcal{T}} \text{shift}(p) \\
=& \; 2\sum_{o \in O-S} f_o + \sum_{p \in \mathcal{S}} (\text{shift}(p) + \text{length}(p)) + 2\sum_{p \in \mathcal{T}} \text{shift}(p) \\
\le& \; 2\left(\sum_{o \in O-S} f_o + \sum_{j \in C} O_j\right)
\end{aligned}$$

where the first inequality follows from Lemmas 5 and 4. This implies that

$$\sum_{s\in S} f_s \le 2\left(\sum_{o\in O-S} f_o + \sum_{j\in C} O_j\right) + \sum_{o\in S\cap O} f_o \le 2\left(\sum_{o\in O} f_o + \sum_{j\in C} O_j\right)$$

which is the statement of Theorem 2 when $S \cap O \neq \phi$.

5 A tight example

Our tight example consists of r facilities in the optimum solution O, r facilities in the locally optimum solution S and rU clients. The facilities are $F = O \cup S$. Since no facility can serve more than U clients, each facility in S and O serves exactly U clients. Our instance has the property that a facility in O and a facility in S share at most one client.

We can view our instance as a set-system—the set of facilities O is the ground set and for every facility $s \in S$ we have a subset X_s of this ground set. $o \in X_s$ iff there is a client which is served by s in the solution S and by o in the solution O. This immediately implies that each element of the ground set is in exactly U sets and that each set is of size exactly U. A third property we require is that two sets have at most one element in common.

We now show how to construct a set system with the properties mentioned above. With every $o \in O$ we associate a distinct point $x^o = (x_1^o, x_2^o, \dots x_U^o)$ in a U-dimensional space where for all i, $x_i^o \in \{1, 2, 3, \dots, U\}$. For every choice of coordinate i, $1 \le i \le U$ we form U^{U-1} sets, each of which contains all points differing only in coordinate i. Thus the total number of sets we form is $r = U^U$ which is the same as the number of points. Each set can be viewed as a line in U-dimensional space. To see that this set system satisfies all the properties note that each line contains U points and each point is on exactly U lines. It also follows from our construction that two distinct lines meet in at most one point.

We now define the facility and the service costs. For a facility $o \in O$, $f_o = 2U$ while for facility $s \in S$, $f_s = 6U - 6$. For a client $j \in N_s^o$, we have $c_{s,j} = 3$ and $c_{o,j} = 1$. All other service costs are given by the metric property.

Lemma 6 *For a client j and facility $s \in S$, the three smallest values that $c_{s,j}$ can have are 3,5 and 11. Similarly, the three smallest values that $c_{o,j}$, $o \in O$ can have are 1,7 and 9.*

Proof A client j can be served at a cost 1 by exactly one facility in O and at a cost 3 by exactly one facility in S. The distance between a facility in O and a facility in S is at least 4. □

Since the service cost of each client in O is 1 and the facility cost of each facility in O is $2U$, we have cost$(O) = 3U^{U+1}$. Similarly, cost$(S) = (3 - 2/U)3U^{U+1}$ and hence cost$(S) = (3 - 2/U)$cost(O). We now need to prove that S is indeed a locally optimum solution with respect to the local search operations of add, delete and swap.

Adding a facility $o \in O$ to the solution S, would incur an opening cost of $2U$. The optimum assignment would reassign only the clients in $N_o(O)$, and all these are assigned to o. The reduction in the service cost due to this is exactly $2U$ which is offset by the increase in the facility cost. Hence the cost of the solution does not improve.

If we delete a facility in the solution S, the solution is no longer feasible since the total capacity of the facilities is now $U^{U+1} - U$ and the number of clients is U^{U+1}.

Now, consider swapping a facility $s \in S$ with a facility $o \in O$. The net decrease in the facility cost is $4U - 6$. To bound the increase in service costs we consider a bipartite graph with the facilities $S \cup \{o\}$ and the clients C forming the two sides of the bipartition. Let E be the edges corresponding to the original assignment of clients to facilities and E' be the edges of the new assignment. The symmetric difference of E and E' is a collection of U edge-disjoint paths between s and o. Let $\mathcal{P}$ be this collection and P be one of these paths. We define the *net-cost* of P as the difference between the costs of the edges of E' and E in P.

Lemma 7 *The two paths in $\mathcal{P}$ with the smallest net-cost have a total net-cost of at least 2. All other paths in $\mathcal{P}$ have net-cost of at least 4.*

Note that the increase in service cost as a result of the swap $\langle s, o\rangle$ equals the total net-cost of the paths in $\mathcal{P}$. The lemma implies that the net-cost of the paths is at least $4(U-2)+2$ which is exactly equal to the decrease in facility cost. Hence, swapping any pair of facilities $s \in S$ and $o \in O$ does not improve the solution.

Proof The edges of E on path P have cost 3. From Lemma 6 it follows that the edge on path P incident to o has cost 1,7 or higher while the remaining edges of $P \cap E'$ have cost 5,9 or higher. Edges on P alternate between sets E and E'. Hence starting from s we can pair consecutive edges of P with the first edge of each pair from E and the other from E'. Note that every pair, except the last, contributes at least 2 to the net-cost of P while the last pair contributes at least -2.

1. If the edge of P incident to o has cost 7 or higher then the last pair contributes at least 4 to the net-cost of P and hence the net-cost of P is at least 4.
2. If any edge of $P \cap E'$ has cost 9 or higher then the corresponding pair contributes at least 6 to the net-cost. Since the last pair contributes at least -2, the net-cost of P is at least 4.

As a consequence of the above we can assume that all edges of $P \cap E'$ have cost 5, except the edge incident to o which has cost 1. This implies that the path P corresponds to a path $S_1, S_2, \dots S_k$ in our set-system where consecutive sets have a common element and S_1 corresponds to facility s while S_k contains the element corresponding to o. Alternatively, in our construction of the set-system, the path P corresponds to a sequence of lines where consecutive lines in the sequence intersect and the first line is the one corresponding to facility s while the last line contains the point corresponding to o. Note that the paths in $\mathcal{P}$ are edge-disjoint but not vertex-disjoint. Hence the sequence of lines corresponding to two paths in $\mathcal{P}$ may have common lines but no pair of consecutive lines can be common in the two sequences. Further, the sequences should end in different lines.

A path P containing k sets, corresponds to a sequence of lines containing k lines and has a net-cost of $2(k-2)$. Hence paths with 4 or more lines have a net-cost at

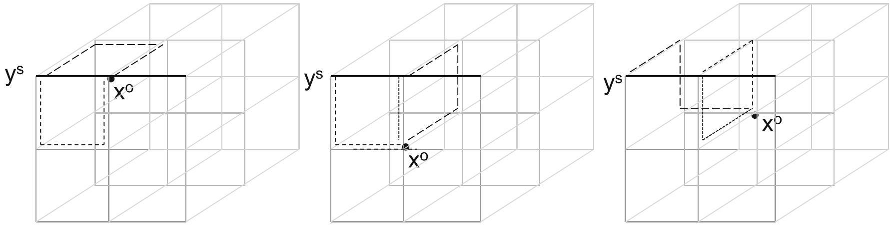

Fig. 4 $U = 3$. The figure shows the three cases corresponding to P_0 having length 1, 2 and 3. P_0 is the *dotted path*, P_1 is the path with small dashes while P_2 is the path with *longer dashes*

least 4 and so to prove the lemma we need to argue that there are at most 2 paths in $\mathcal{P}$ having less than 4 lines. Let P_0, P_1 be the two paths with the smallest lengths with P_0 being the smallest.

1. If P_0 has length 1 then the line corresponding to s, say y^s, contains the point corresponding to o, say x^o. From our construction it follows that any other sequence of lines which starts with y^s and ends with a line containing x^o which is different from y^s must contain at least 4 lines (including line y^s). Hence path P_1 has a net-cost at least 4. Thus the total net-cost of paths P_0 and P_1 is at least 2 (see Fig. 4).
2. If P_0 has length 2 then the line y^s and the point x^o, have identical values for $U - 2$ coordinates. Let y^a be the line in P_0 containing x^o. Once again, from our construction it follows that any other sequence of lines which starts with y^s and ends with a line containing x^o which is different from y^a must contain at least 3 lines. Hence P_1 has length at least 3 and so the total net-cost of paths P_0 and P_1 is at least 2. Further the other paths of $\mathcal{P}$ would end with lines which are in dimensions other than the last lines of P_0, P_1 and so the length of these paths is at least 4 (see Fig. 4).
3. If P_0 has length 3 then y^s and x^o have identical values for $U - 3$ coordinates. In this case, the net-cost of paths P_0, P_1 is at least 2 and the other paths of $\mathcal{P}$ have at least 4 lines (see Fig. 4).

□

6 Conclusions

While the local search algorithm for capacitated facility location is easy to specify, the analysis, even for the case of uniform capacities, can be quite involved. Analyzing the more general case of non-uniform capacities would be quite a challenge. This suggests that one should explore other, non-LP, non-local-search approaches to capacitated facility location.

References

1. Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., Pandit, V.: Local search heuristics for k-median and facility location problems. SIAM J. Comput. **33**(3), 544–562 (2004)

2. Charikar, M., Guha, S.: Improved combinatorial algorithms for the facility location and k-median problems. In: Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science, pp. 378–388 (1999)
3. Chudak, F., Williamson, D.P.: Improved approximation algorithms for capacitated facility location problems. Math. Program. **102**(2), 207–222 (2005)
4. Korupolu, M.R., Plaxton, C.G., Rajaraman, R.: Analysis of a local search heuristic for facility location problems. J. Algorithms **37**(1), 146–188 (2000)
5. Mahdian, M., Pál, M.: Universal facility location. In: Proceedings of the 11th Annual European Symposium on Algorithms, volume 2832 of Lecture Notes in Computer Science, pp. 409–421 (2003)
6. Mahdian, M., Ye, Y., Zhang, J.: A 2-approximation algorithm for the soft-capacitated facility location problem. In: Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization, volume 2764 of Lecture Notes in Computer Science, pp. 129–140 (2003)
7. Pál, M., Tardos, É., Wexler, T.: Facility location with nonuniform hard capacities. In: FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science, p. 329. IEEE Computer Society, Washington, DC, USA (2001)
8. Zhang, J., Chen, B., Ye, Y.: A multiexchange local search algorithm for the capacitated facility location problem. Math. Oper. Res. **30**(2), 389–403 (2005)